Assignment 6 – Semantic Segmentation (ACDC Dataset)

Weight: 5% (2% In-lab + 3% Take-home)

Overview

In this assignment, you will implement and analyze a **semantic segmentation model** (e.g., **U-Net**, **DeepLabV3**, or an equivalent CNN-based architecture) on the **Automated Cardiac Diagnosis Challenge (ACDC)** dataset. The ACDC dataset contains short-axis cardiac MR images annotated with pixel-wise segmentation masks for the **Left Ventricle (LV)**, **Right Ventricle (RV)**, and **Myocardium (MYO)**. Your goal is to predict accurate segmentation masks for these anatomical structures and evaluate model performance using standard metrics: **F1 Score, Intersection over Union (IoU), Sensitivity, Specificity, and Precision**.

You will train or fine-tune a segmentation model using appropriate augmentations and hyperparameter tuning. Quantitative and qualitative analyses are both required.

Model Size Constraint: Your final model (weights + architecture) must not exceed **200 MB**. Submissions exceeding this limit or unable to run on a Colab GPU will not receive credit.

Dataset & Environment

- **Dataset:** ACDC (Automated Cardiac Diagnosis Challenge)
- **Input:** 2D short-axis MRI slices (grayscale or pseudo-RGB)
- **Classes:** 4 total – Background (0), LV (1), RV (2), Myocardium (3)
- **Device:** GPU (Colab or local)
- **Libraries:** PyTorch + MONAI / torchvision / segmentation_models.pytorch
- **Output:** Segmentation maps with 4 classes, same spatial size as input

Part A — In-Lab (2% of course grade)

Goal: Train a UNET model on ACDC and demonstrate correct pixel-wise predictions. The code for UNET and its training are provided.

A.1 Tasks

1. **Utilies:** Load a subset of ACDC (e.g., ∼20 patients). Visualize MR slices with their corresponding ground-truth masks.

2. **Model Setup:** Train provided U-Net

3. **Inference:** Run inference on the validation subset and overlay predicted masks on the original MR images.

4. **Metrics:** Compute the following metrics for the test set:

    - **F1 Score (Dice coefficient)**
    - **IoU (Intersection over Union)**
    - **Sensitivity (Recall)**
    - **Specificity**
    - **Precision**

A.2 Expected Performance (In-class)

Baseline target: F1 Score ≥ 0.90, IoU ≥ 0.90, Sensitivity ≥ 0.90, Specificity ≥ 0.90, and Precision ≥ 0.90.

Part B — Take-Home (3% of course grade)

Goal: Fine-tune a segmentation model on the full ACDC dataset, apply data augmentations, and perform quantitative and qualitative analysis.

B.1 Tasks

1. **Transformations:** Train or fine-tune on the full ACDC dataset using suitable augmentations (rotation, flipping, intensity shifts, elastic deformations).

2. **Evaluation:** Compute the metrics (F1, IoU, Sensitivity, Specificity, Precision)

3. **Visualization:** Display 5–10 qualitative examples comparing predicted and ground-truth masks.

4. **Report:** Provide a brief discussion covering:

    - The effect of data augmentation on model generalization,
    - Metric trade-offs (e.g., high sensitivity vs. low precision),
    - Common failure modes (e.g., poor delineation of thin myocardium),
    - Comparison to the detection-only results from Assignment 5.

Deliverables Checklist

- **Notebook:** `A6.ipynb` (runs top-to-bottom on a clean runtime)
- **Report (PDF):** $\leq$ 2 pages with metrics, tables, and qualitative results
- **Trained weights for UNET and your custom model:** `.pt` or `.pth` file $\leq$ 200 MB total

Marking Breakdown (5% total)

Part A (In-Lab — 2%)

Requirement	Value (%)	Notes
Dataset loaded & visualized	0.5	Visualize 20 samples (raw images and ground truth masks)
Metric computation	0.5	Correctly implement F1, IoU, Sensitivity, Specificity, and Precision
UNET Training	1.0	Train the UNET model so that final training loss is less than 0.20 and all the metrics you implemented above are greater than 0.90

Part B (Take-Home — 3%)

Requirement	Value (%)	Notes
Custom Model	0.5	Defined a custom model of your choice. Model should be reasonable (not too small or too big). Look at course notebooks for help.
Metric Visualization	0.25	All metrics are visualized in their own plots (you'll need to change the train_per_epoch function to track metrics during training). You need to compare the metrics between the UNET model and your custom model.
Image and joint transformations	0.25	Pick a reasonable set of transformations to use during training
Report documentation	2.0	Discuss the performance of your custom model. Discuss why you chose the transformations that you did. Include ALL plots in the report.

** CAUTION: If you do not include ALL plots in the report, you will lose the full 2

Recommended Resources

- ACDC Challenge Homepage
- MONAI Tutorials (ACDC Segmentation Examples)
- Torchvision Segmentation Tutorial
- PyTorch U-Net Tutorial
- DeepLabV3+ PyTorch Implementation